

Perbandingan Algoritma Elias Gamma Code Dan Punctured Elias Code Dalam Mengkompresi File Audio MP3

Arfan Heri

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: arfanheri711@gmail.com
Email Penulis Korespondensi: arfanheri711@gmail.com

Abstrak—Di era digital saat ini penggunaan teknologi kompresi data sangat penting disebabkan pemanfaatan data dalam bentuk elektronik lebih cepat untuk melakukan pertukaran informasi. Ukuran file yang besar membutuhkan banyak ruang penyimpanan. File audio berekstensi MP3 memiliki ukuran yang besar dibanding dengan jenis file Video yang berekstensi lain. Dikarenakan ukuran file berekstensi .MP3 terlalu besar mengakibatkan proses pengiriman data memerlukan waktu yang cukup lama. Salah satu teknik yang diperlukan untuk masalah ini adalah melakukan kompresi. Kompresi adalah proses pengubahan ukuran data ke ukuran yang lebih kecil. Algoritma yang digunakan pada penelitian ini adalah algoritma Elias Gamma Code dan Punctured Elias Code yang memiliki kelebihan dan kekurangan masing-masing. Setelah kedua algoritma dikompres, kemudian dilakukan perbandingan dari kedua algoritma tersebut menggunakan metode lossless compression. Metode perbandingan lossless compression merupakan salah satu metode yang digunakan untuk menentukan urutan prioritas alternatif keputusan dengan beberapa kriteria. Hasil akhir dari proses perbandingan dengan metode lossless compression didapatkan hasil 74,2% yang merupakan algoritma Punctured Elias Code. Sehingga algoritma tersebut menjadi algoritma tercepat dan efektif dalam melakukan proses kompresi file audio, hal ini dikarenakan semakin kecil total nilai yang diperoleh maka semakin sedikit jumlah usaha yang dilakukan oleh algoritma tersebut dalam melakukan kompresi.

Kata Kunci: Kompresi, File Audio, Perbandingan, Elias Gamma Code, Punctured Elias Co

Abstract—In today's digital era, the use of data compression technology is very important because the utilization of data in electronic form is faster for exchanging information. Large file sizes require a lot of storage space. MP3-extended audio files have a large size compared to other types of video files. Because the size of the MP3 file is too large, the process of sending data takes a long time. One of the techniques needed for this problem is compression. Compression is the process of resizing data to a smaller size. The algorithms used in this research are the Elias Gamma Code and Punctured Elias Code algorithms, which have their own advantages and disadvantages. After the two algorithms are compressed, a comparison of the two algorithms is made using the lossless compression method. The lossless compression comparison method is one of the methods used to determine the priority order of alternative decisions based on several criteria. The final result of the comparison process with the lossless compression method obtained a result of 74.2%, which is the Punctured Elias Code algorithm. So that the algorithm becomes the fastest and most effective algorithm in compressing audio files, this is because the smaller the total value obtained, the less effort made by the algorithm in compression.

Keywords: Compression, Audio Files, Comparison, Elias Gamma Code, Punctured Elias Co

1. PENDAHULUAN

Dalam perkembangan saat ini, komputer digunakan untuk membantu dan mempercepat kinerja manusia. Dalam hal komunikasi data, pesan yang dikirim seringkali ukurannya sangat besar sehingga waktu pengirimannya lama. Begitu juga dalam penyimpanan data, arsip atau file yang berukuran besar membutuhkan ruang penyimpanan yang besar. Kedua masalah tersebut dapat diatasi dengan mengkodekan pesan atau isi arsip sesingkat mungkin, sehingga waktu pengiriman data juga relatif cepat, ruang penyimpanan yang dibutuhkan juga sedikit, cara pengkodean seperti ini disebut pemampatkan atau kompresi data[1].

Kompresi adalah suatu proses pengkodean informasi menggunakan bit-bit atau information-bearing pada unit lainnya yang lebih rendah jika dibandingkan dengan representasi data yang tidak terkodekan menggunakan sistem encoding tertentu. Pada pihak pengirim data harus menggunakan algoritma kompresi data yang sudah distandarisasi dan pada pihak penerima pun harus menggunakan suatu teknik dekompresi yang sama dengan yang dikirimkan oleh pengirim hal ini dimaksudkan supaya data yang diterima oleh penerima dapat dibaca kembali dengan benar[2].

File audio yang digunakan dalam penelitian ini adalah sebuah file audio yang berekstensi MP3 yang merupakan salah satu format file audio yang sering digunakan dalam pengunduhan dan streaming audio dari media internet. Dalam mengkompresinya format MP3 bersifat flaksibel karena dapat menyimpan suara music. Pada umumnya file audio memiliki ukuran yang lebih besar dibandingkan file teks, file gambar sehingga untuk mengurangi ukurannya dapat dilakukan kompresi. Seperti teknik kompresi pada umumnya ada dua jenis kompresi data yaitu lossy dan lossless yang kompresi file, file dikembalikan seperti data asli tanpa mengalami perubahan atau data tidak ada yang hilang.

Pada saat ingin mengkompresi suatu data. Disini penulis akan membandingkan kinerja dari algoritma Elias Gamma Code dan algoritma Punctured Code. Algoritma Elias Gamma Code yang merupakan algoritma ciptaan Peter Elias adalah salah satu algoritma kompresi yang dapat mengkodekan bilangan bulat positif[3]. Algoritma ini bersifat Lossless yang di mana menggunakan teknik pengurangan kapasitas sebuah file tanpa harus mempengaruhi file aslinya yang kemudian hasil kompresi tersebut dapat didekompresi atau dikembalikan bentuk file awal secara utuh[4]. Sedangkan algoritma Punctured code merupakan suatu metode yang dirancang oleh Peter Fenwick dalam sebuah percobaan untuk meningkatkan performa the Burrows-wheeler Transfor yang bersifat lossless[1].

Pada penelitian ditahun 2019 yang dilakukan oleh Dian Pratiwi dkk telah mampu mengkompresi file teks menghasilkan ratio of compression sebesar 1,66 bit untuk algoritma Fixed Length Binary Encoding, sedangkan algoritma Elias Gamma Code sebesar 1,62 bit. Untuk rata-rata compression ratio algoritma Fixed Length Binary Encoding sebesar 60,9% sementara itu algoritma Elias Gamma Code mendapatkan rasio sebesar 62,20%. Dan untuk nilai rata-rata Time kompresi pada algoritma Fixed Length Binary Encoding sebesar 16 ms sedangkan algoritma Elias Gamma Code sebesar 21 ms[4]. Pada tahun 2019 penelitian yang dilakukan oleh Hikka Sartika dkk dalam penggunaan algoritma Elias Gamma Code pada kompresi record database aplikasi rangkuman pengetahuan umum lengkap dengan sampel uji sebesar 64 bit dan memperoleh hasil ratio of compression sebesar 2 bit, untuk compression ratio sebesar 50% dengan redundancy sebesar 50%. Berdasarkan hasil penelitian sebelumnya yang dilakukan oleh Dedek Andri Yansyah pada tahun 2015 telah berhasil mengkompresi file text dengan menggunakan metode punctured elias codes yang berektensi doc hasil output dekompresi telah diatur dengan format perataan text justify dan jenis font times new roman dengan ukuran font Penulis Memilih metode Punctured Elias Code. berdasarkan dari penelitian sebelumnya yang dilakukan oleh Desi Husni dalam skripsinya yang berjudul” Implementasi dan Analisis Algoritma Kompresi Punctured Elias Code dan Ternary Comma Code pada File Doc” Berdasarkan penelitian ini, akan dilakukan dalam aplikasi yaitu dengan cara menerapkan algoritma Punctured Elias Code pada kompresi aplikasi. Untuk menerapkannya terlebih dahulu mencari nilai biner aplikasi pada aplikasi binary viewer, setelah mendapatkan nilai biner maka biner tersebut disusun untuk mencari frekunsinya, setelah nilai frekuensi tersusun selanjutnya melakukan kompresi. Kompresi dengan menggunakan metode algoritmaPunctured Elias Code dapat memperkecilkan ukuran aplikasi yang besar menjadi lebih kecil setelah mendapatkan hasil kompresi maka tahap selanjutnya mengembalikan hasil aplikasi yang sudah dikompresi mejadi aplikasi asli yaitu dengan cara mendekompresi file tersebut. File yang didekompresi akan kembali ke aplikasi sebelumnya. Cara kerja algoritma Punctured Elias Code pada kompresi dan dekompresi aplikasi yaitu dengan menginputkan aplikasi setelah aplikasi diinput maka nilai biner aplikasi terbaca dan nilai biner tersebut diurutkan berdasarkan frekunesi.

Berdasarkan pengujian kompresi terhadap file doc, dapat disimpulkan bahwa kompresi dengan metode Ternary Coma code waktu kompresi dan dekompresinya yang dihasilkan jauh lebih besar dari pada algoritma Punctured Elias Code. Berdasarkan latar belakang masalah diatas, penulis bermaksud untuk menguraikan tahap-tahap pemberian solusi yang akan jelas dalam penelitian ini dengan berjudul “Perbandingan Algoritma Elias Gamma Code Dan Punctured Elias Code Dalam Mengkompresi File Audio (MP3)”.

2. METODE PENELITIAN

2.1 Kompresi Data

Kompresi melibatkan pengurangan ukuran file yang besar menjadi lebih kecil, dengan tujuan menghemat ruang penyimpanan. Dalam konteks audio, ini mencakup metode untuk mengurangi ukuran file audio dengan mengurangi setiap bit yang ada dalam data audio. Kompresi data, di sisi lain, berfokus pada mengurangi ukuran data tanpa merusak kualitas atau dimensi dari setiap file individual. Proses kompresi data didasarkan pada pengamatan bahwa jenis data tertentu sering memiliki pola berulang dalam komponennya. Proses kompresi data juga melibatkan langkah encoding yang bertujuan untuk menghilangkan elemen berulang tersebut, menghasilkan ukuran data yang lebih kecil[4], [5].

2.2 Algoritma Elias Gamma Codes

Algoritma elias gamma code dibuat oleh Peter Elias. Tabel kode elias gamma, elias dibuat dengan menambahpanjang kode dalam Unary (u). Kode berikutnya, E_y ditambahkan pada panjang kode (M) dalam biner (β)[6][7]. Dengan demikian, elias gamma code, yang juga untuk bilangan bulat positif, sedikit lebih kompleks untuk dibangun[8]. Aturan untuk mengkodekan sebuah bilangan dengan menggunakan Elias Gamma[9] adalah sebagai berikut:

1. Rubah bilangan kode dalam bentuk biner
2. Kurang 1 dari jumlah bit yang dipilih pada langkah pertama dan tambahkan sesuai dengan banyaknya bilangan Nol. Proses yang ekuivalen untuk menyatakan proses yang pada point nomor dua adalah sebagai berikut :
 - a. Pisahkan integer menjadi pangkat 2 tertinggi (2^N) yang dapat dan ditampung sisakan digit biner N dari integer tersebut.
 - b. Kodekan N dalam bentuk unary, jika N adalah nol maka diikuti oleh satu.
 - c. Tambahkan sisa digit biner N yang telah dihasilkan
3. Tentukan nilai N untuk pangkat yang paling mendekati nilai n yang ditulis sebagai $\beta(n)$. Nilai ini disebut sebagai unary code, dimana jumlah nilai N ditulis menjadi angka 0 dan diakhiri dengan angka 1.
4. Dapat nilai L dengan mengurangi nilai n nilai (2^N) nilai yang didapati diubah menjadi bilangan biner pembentukan kode elias gamma
5. Langkah berikut adalah mengkonvresi krakter sesuai dengan Kode Elias Gamma, sebelum dikonversi terlebih dahulu pemeriksaan terhadap panjang string bit dengan langkah sbagai berikut :
 - a. Jika sisa bagi panjang string bit terhadap 8 adalah 0 maka tambahkan 0000001. Nyata dengan bit akhir.

- b. Jika sisa bagi panjang string bit terhadap 8 adalah n (1,2,3,4,5,6,7,) maka tambahkan 0 sebanyak $7 - n + "1"$ di akhir string bit. Nyatanya dengan L lalu tambahkan bilangan biner dari $9 - n$. Nyatanya dengan bit akhir.

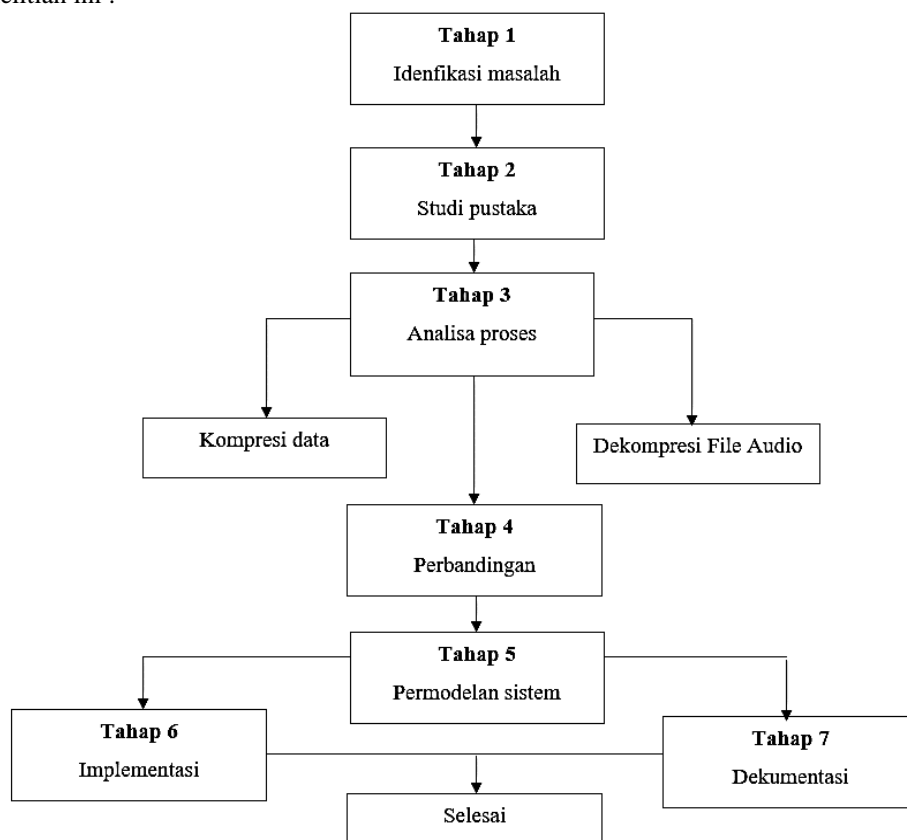
2.3 Algoritma Puncture Elias Code

Puncture elias code bilangan intergar dirancang oleh Petter Finwick dalam sebuah percobaan unruk meningkatkan performa. Terdiri data yang asli ditambah sejumlah bilangan dari check bits. Jika beberapa check bits dihilangkan, untuk mempersingkat serangkaian kode itu, hasil kode ditujukan sebagai puncturead[10]. Cara untuk membangun kode puncturead elias code adalah:

1. Ambil bilangan biner dari n ,
2. Reversed (balikan bit-bitnya), dan siapkan flag untuk menunjukan jumlah bit yang bernilai 1 didalam n
3. Untuk setiap bit didalam n kita siap flag dengan 0
4. Gabungkan flag bilangan biner yang sudah di balikan (reversed)

2.4 Tahapan Penelitian

Kerangka penelitian terdiri dari beberapa langkah yang disusun secara sitematis dan nantinya akan dihitung secara bersama. Kerangka kerja penelitian ini bertujuan untuk memudahkan pelaksanaan penelitian. Berikut adalah kerangka kerja dalam penelitian ini :



Gambar 1. Kerangka Kerja Penelitian

Dibawah ini merupakan tahapan sederhana proses pengumpulan data dari penelitian yang dilakukan.

1. Identifikasi masalah
Pada langkah ini, merupakan metode yang diterapkan oleh penulis untuk melakukan prediksi dan estimasi. serta menggambarkan sebuah sumber masalah terkait transfer file yang kerja terhambat, yaitu adalah file audio yang berekstensi MP3 yang bahkan file tersebut tidak bias muat lagi dimemori penyimpanan. Untuk itu penulis menggunakan teknik kompresi agar menghemat ruang memori penyimpanan.
2. Studi pustaka
Pada langkah ini, dilakukan proses pengumpulan data yang terkait dengan algoritma Elias Gamma Code dan Punctured Elias Code. Peneliti menghimpun informasi sebagai acuan dari berbagai sumber seperti buku, jurnal, dan situs internet yang relevan dengan topik penelitian.
3. Analisa Proses Kompresi Dan Dekompresi Pada File audio
Pada tahap ini merupakan cara melakukan proses analisa untuk menentukan proses kompresi dan dekompresi dalam konteks penggunaan algoritma Elias Gamma Code dan Punctured Elias Code pada file audio, keduanya memiliki tujuan yang serupa, yaitu mengurangi ukuran data. Dalam kedua algoritma ini, dapat dilakukan perhitungan codeword untuk setiap data (n) yang ada. Selanjutnya, untuk analisis proses dekompresi, langkah ini berfokus pada

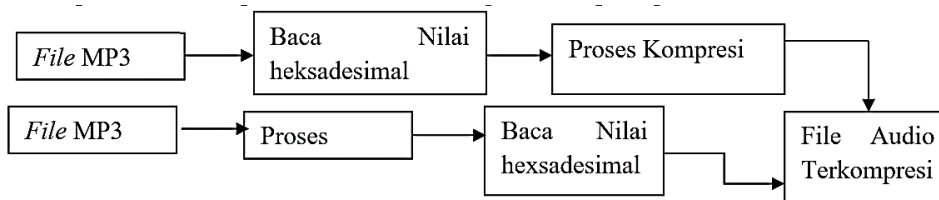
mengembalikan file audio yang telah terkompresi ke bentuk aslinya sebelum mengalami kompresi melalui penggunaan algoritma Elias Gamma Code dan Punctured Elias Code.

4. Perbandingan
 Pada langkah ini, dilaksanakan untuk mengevaluasi dan membandingkan metode kompresi algoritma Elias Gamma Code dan Punctured Elias Code dalam konteks file audio. Data yang terdapat pada dataset aplikasi digunakan untuk melaksanakan perbandingan antara kedua algoritma ini. Tujuannya adalah untuk menentukan sistem kompresi mana yang lebih efektif dan sesuai dengan implementasi yang dijalankan.
5. Permodelan Sistem
 Pada tahap ini merupakan penyederhanaan dari sebuah elemen dan komponen yang sangat kompleks untuk memudahkan pemahaman dan informasi yang dibutuhkan. Didalam sistem tersebut juga terdapat pengujian terhadap sistem yang telah dibangun sehingga dalam pengujian tersebut dapat diketahui apakah sistem telah berkerja dengan benar dalam perbandingan proses kompresi dan dekompresi pada file audio berekstensi MP3
6. Implementasi
 Pada tahap ini dilakukan pengimplementasian file audio berekstensi MP3. Menggunakan algoritma Elias Gamma Code dan Punctured Elias Code
7. Dekumentasi
 Pada ini dilakuan dengan mendokumentasi hasil analisa dan pengujian secara tertulis dalam bentuk laporan sikripsi

3. HASIL DAN PEMBAHASAN

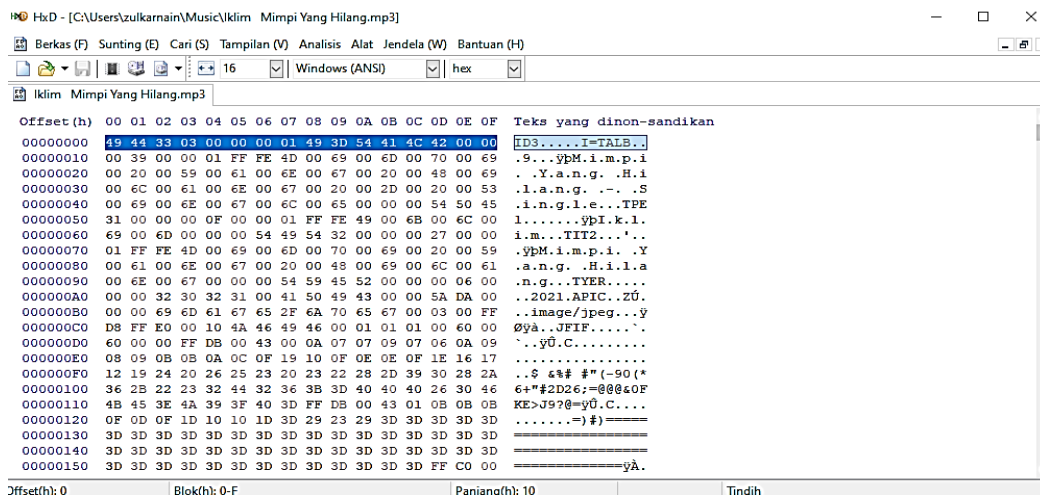
3.1 Analisa

Dalam konteks penelitian ini, algoritma Elias Gamma Code dan Punctured Elias Code digunakan untuk melakukan kompresi pada file MP3. Perbandingan antara kedua algoritma ini dilakukan untuk mengidentifikasi algoritma yang lebih akurat dalam mengurangi ukuran file MP3. Proses kompresi pada file MP3 bertujuan untuk mengurangi kebutuhan ruang penyimpanan dan meningkatkan kecepatan pengiriman file. Fokus utama dalam penelitian ini adalah membandingkan parameter kompresi seperti Rasio Kompresi, Compression Ratio, redundansi, penghematan ruang, serta proses dekompresi dari algoritma Elias Gamma Code dan Punctured Elias Code. Langkah ini bertujuan untuk memahami lebih dalam tentang bagaimana proses kompresi dan dekompresi dilakukan pada suatu file MP3. dapat dilihat pada gambar dibawah ini.



Gambar 2. Prosedur Kompresi Dan Dekompresi File

Dalam tahap kompresi, dua algoritma yang digunakan adalah algoritma Elias Gamma Code dan Punctured Elias Code. Sebelum melakukan kompresi pada file MP3, tahap awal melibatkan pembacaan file MP3 untuk mendapatkan representasi nilai heksadesimal menggunakan aplikasi HxD. Contoh hasil dari file MP3 yang telah dikompresi dan dekompresi dapat dilihat dalam gambar 3 berikut ini:



Gambar 3. Isi File MP3 dari hasil HxD

Dari ilustrasi di atas, terambil contoh sampel nilai heksadesimal dari file MP3. Dalam rangka perhitungan manual, hanya sejumlah 16 karakter nilai heksadesimal yang akan diambil dari sisi kiri hingga kanan file MP3. Dilakukan langkah pembacaan isi dari file tersebut. Berikut adalah contoh nilai heksadesimal dari file MP3 ini: 49,44,33,03,00,00,00,01,49,3D,54,41,4C,42,00,00 Data nilai ini diatur dalam tabel untuk memungkinkan analisis frekuensi. Frekuensi dihitung dengan menghitung jumlah kemunculan nilai yang sama dalam setiap entri. Kemudian, langkah selanjutnya melibatkan penyusunan karakter berdasarkan frekuensi terbesar hingga frekuensi terkecil. Pengaturan urutan nilai ini tercantum dalam tabel 1 berikut:

Tabel 1. Nilai Bit MP3 Sampel

Nilai		Bit	Frek	Bit * Frek
Hexa	Biner			
00	00000000	8	5	40
49	01001001	8	2	16
33	00110011	8	1	8
03	00000011	8	1	8
01	00000001	8	1	8
3D	00111101	8	1	8
54	01010100	8	1	8
41	01000001	8	1	8
4C	01001100	8	1	8
42	01000010	8	1	8
Total Bit				128

Dari tabel 1 diketahui bahwa setiap nilai heksadesimal (karakter) memiliki panjang 8 bit dalam bentuk bilangan biner. Oleh karena itu, jika terdapat 16 bilangan heksadesimal, maka total panjang nilai binernya mencapai 128 bit.

3.2 Penerapan Elias Gamma Code

Kompresi nilai pada sample kode Elias Gamma Code yang didapat pada table 2 dibawah. Adapun proses kompresi file MP3 sample dapat dilihat pada table berikut.

Tabel 2. Kompresi Nilai File Audio MP3 Sample Dengan Elias Gamma Code

N	Nilai Hexa	Elias Gamma Code	Bit	Frek	Bit x Frek
1	00	1	1	5	5
2	49	010	3	2	6
3	44	011	3	1	3
4	33	00100	5	1	5
5	03	00101	5	1	5
6	01	00110	5	1	5
7	3D	00111	5	1	5
8	54	0001000	7	1	7
9	41	0001001	7	1	7
10	4C	0001010	7	1	7
11	42	0001011	7	1	7
Total Bit					62

Presentase pengurangan ukuran data yang telah mengalami kompresi dapat diidentifikasi melalui perbandingan antara ukuran data setelah dikompresi dengan ukuran data sebelum mengalami kompresi. Dari data ini dilakukan perhitungan kinerja kompresi yaitu :

1. Ratio Compression (Rc)

$$Rc = \frac{\text{Ukuran data Sebelum Dikompresi}}{\text{Ukuran data sesudah Dikompresi}}$$

$$Rc = \frac{128}{62}$$

$$Rc = 2,0$$

2. Compression Ratio (Cr)

$$Cr = \frac{\text{Ukuran data Sebelum Dikompresi}}{\text{Ukuran data sesudah Dikompresi}} * 100\%$$

$$Cr = \frac{62}{128}$$

$$Cr = 48,4\%$$

3. Redudancy (Rd)

$$Rd = \frac{\text{Ukuran sebelum dikompresi} - \text{ukuran sesudah dikompresi}}{\text{Ukuran data sebelum Dikompresi}} * 100\%$$

$$Rd = \frac{128-62}{128} * 100\%$$

$$Rd = 51,0 \%$$

4. Space Saving (Ss)

$$Ss = 100\% - Cr$$

$$Ss = 100\% - 48,4\%$$

$$Ss = 51,6\%$$

Pada analisa proses dekompresi file MP3 menggunakan algoritma Elias Gamma Code dilakukan proses memasukan hasil file kompresi yaitu: 0,5,BO,CB,49,12,34,0,36,2,46,81,23,70,E5,D3,B5,21,2F,A,1. Hasil string bit dari algoritma Elias Gamma Code yang digunakan sebagai nilai file diubah menjadi representasi dalam bentuk biner. Keseluruhan bit hasil dari proses kompresi dapat dilihat.

Tabel 3. Nilai File Dan Kode Elias Gamma Code

N	Nilai Hexa	Kode Elias Gamma Code
1	00	1
2	49	010
3	33	011
4	03	00100
5	00	00101
6	00	00110
7	01	00111
8	3D	0001000
9	54	0001001
10	4C	0001010
11	42	0001011

Dari hasil dekompresi di atas, didapat nilai hexadesimal awal sebelum kompresi yaitu 49, 33, 03, 00, 00, 01, 3D, 54, 4C, 42, 00.

3.3 Penerapan Algoritma Punctured Elias Code

Setelah melakukan perhitungan terhadap data sebelum dikompresi dari tabel di atas, langkah berikutnya adalah melaksanakan proses kompresi dengan menerapkan algoritma Punctured Elias Code.

Tabel 4. Nilai Hexadesimal Setelah Dikompresi Mengguna Algoritma Puncturead Elias Code

N	Nilai Hexa	Puncturead Elias Code	Bit	Frek	Bit x Frek
1	00	1	1	5	5
2	49	101	3	2	6
3	44	1001	4	1	4
4	33	11011	5	1	5
5	03	10001	5	1	5
6	01	110101	6	1	7
7	3D	1110111	7	1	8
8	54	100001	6	1	7
Total bit					46

Persentase pengurangan ukuran data yang telah dikompresi dapat dilihat melalui hasil perbandingan antara ukuran data setelah dikompresi dan ukuran data sebelum dikompresi. Dengan menggunakan data ini, dapat dihitung kinerja kompresinya sebagai berikut:

1. Ratio compression (Rd)

$$Rc = \frac{\text{Ukuran data Sebelum Dikompresi}}{\text{Ukuran data sesudah Dikompresi}}$$

$$Rc = \frac{128}{46}$$

$$Rc = 2,7$$

2. Compression Ratio (Cr)

$$Cr = \frac{\text{Ukuran data Sesudah Dikompresi}}{\text{Ukuran data Sebelum Dikompresi}} * 100\%$$

$$Cr = \frac{46}{128}$$

$$Cr = 35,9\%$$

3. Redudancy (Rd)

$$Rd = \frac{\text{Ukuran sebelum dikompresi} - \text{ukuran sesudah dikompresi}}{\text{Ukuran data sebelum Dikompresi}} * 100\%$$

$$Rd = \frac{128-46}{46} * 100\%$$

$$Rd = 78,26 \%$$

4. Space saving

$$Ss = 100\% - Cr$$

$$Ss = 100\% - 78,26\%$$

$$Ss = 77,26\%$$

Hasil proses dekomposisi file MP3 menggunakan Algoritma Punctured Elias Code ialah :

Tabel 5. Dekomposisi Nilai Hexadesimal Sampel Audio

Urutan Nilai	Puncturead Elias Code	Nilai hexa	Biner
1	10	00	00000000
2	1010	49	00110001
3	1001	44	00101100
4	11011	33	00100001
5	10001	03	00000011
6	110101	01	00000001
7	1110111	3D	00111101
8	100001	54	00110110

3.4 Analisa Hasil Perbandingan Algoritma

Perbandingan antara algoritma Elias Gamma Code dan Punctured Elias Code berfokus pada evaluasi dari Ratio Compression (Rc), Compression Ratio (Cr), Redundancy (Rd), dan Space Saving yang dihasilkan oleh kedua algoritma. Dalam proses ini, perhitungan dilakukan menggunakan kedua algoritma, yaitu Elias Gamma Code dan Punctured Elias Code, untuk mendapatkan nilai dari Ratio Compression (Rc), Compression Ratio (Cr), Redundancy, dan Space Saving. Setelah diperoleh hasil dari kedua algoritma, dilakukan perbandingan untuk menentukan algoritma mana yang lebih efektif dalam melakukan proses kompresi pada file audio MP3. Hasil perbandingan ini dapat dilihat dalam tabel 6.

Table 6. Nilai Perbandingan

No	Algoritma	Ratio Compression (Rc)	Compression Ratio (Cr)	Redudancy (Rd)	Space Saving (Ss)
1	Elias Gamma Code	2,0%	48,4%	51,0%	51,6%
2	Punctured Elias Code	2,7%	35,9%	78,26%	77,2%

Tabel 7. Hasil Nilai Perbandingan

Nama File	Ukuran sebelum dikompres	Ukuran sesudah dikompres
Iklim – mimpi yang hilang	5.02 MB	2,0%
Nike ardila- mama aku ingin pulang	5.21 MB	2,7%
Mahalini – Sial	3.72 MB	35,9%

Dari penjelasan yang ada pada tabel di atas, dapat disimpulkan bahwa algoritma Elias Gamma Code lebih efektif dalam melakukan kompresi file. audio MP3. Kerena nilai persennya Rc 2,0%, Cr 48,4%, Rd 51,0% memiliki space

saving 51,6% yang dibandingkan dengan algoritma Puncturead Elias Code dan bisa jug dibandingkan dari ukuran Rc 2,7% Cr 35,9%, Rc 78,26% dan Space saving 77,2%.

4. KESIMPULAN

Beberapa kesimpulan dapat diambil setelah mengevaluasi bab-bab sebelumnya, kompresi file audio dalam format MP3 dapat dijalankan dengan mengonversinya ke dalam format heksadesimal, yang memungkinkan untuk melakukan perhitungan bilangan heksadesimal. Bilangan ini kemudian diproses menggunakan algoritma Elias Gamma Code dan Punctured Elias Code, menghasilkan nilai baru yang merubah ukuran file audio MP3. Setelah melakukan perhitungan, kedua algoritma Elias Gamma Code dan Punctured Elias Code dapat dibandingkan. Skala perbandingan meliputi Ratio Compression, Compression Ratio, Redundancy, dan Space Saving. Terlihat bahwa algoritma Elias Gamma Code menghasilkan nilai yang lebih kecil dibandingkan dengan algoritma Punctured Elias Code. Nilai-nilai seperti Ratio Compression (Rc), Compression Ratio (Cr), Redundancy (Rd), dan Space Saving dapat dihitung dengan mengubah nilai awal dari file MP3 menjadi nilai heksadesimal. Untuk algoritma Elias Gamma Code, ditemukan nilai Ratio Compression (Rc) sebesar 2,0%, Compression Ratio (Cr) sebesar 48,4%, Redundancy (Rd) sebesar 51,0%, dan Space Saving sebesar 51,6%. Sementara itu, untuk algoritma Punctured Elias Code, ditemukan nilai Ratio Compression (Cr) sebesar 2,7%, Compression Ratio (Cr) sebesar 35,9%, Redundancy (Rd) sebesar 78,26%, dan Space Saving sebesar 77,2%.

REFERENCES

- [1] A. Pradana, "Analisa Perbandingan Algoritma Elias Gamma Code Dan Algoritma Goldbach Code Pada Kompresi File Dokumen," vol. 6, no. November, pp. 345–356, 2022, doi: 10.30865/komik.v6i1.5741.
- [2] H. Sinaga, P. Sihombing, and H. Handrizal, "Perbandingan Algoritma Huffman Dan Run Length Encoding Untuk Kompresi File Audio," *Talenta Conference Series: Science and Technology (ST)*, vol. 1, no. 1, pp. 010–015, 2018, doi: 10.32734/st.v1i1.183.
- [3] H. Sartika and T. Zebua, "PERANCANGAN DAN IMPLEMENTASI ALGORITMA ELIAS GAMMA CODE UNTUK MENGKOMPRESI RECORD DATABASE PADA APLIKASI RANGKUMAN PENGETAHUAN UMUM LENGKAP," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 259–265, Nov. 2019, doi: 10.30865/komik.v3i1.1600.
- [4] D. Pratiwi and T. Zebua, "Analisis Perbandingan Kinerja Algoritma Fixed Length Binary Encoding Dan Algoritma Elias Gamma Code Dalam Kompresi File Teks," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 424–430, 2019, doi: 10.30865/komik.v3i1.1623.
- [5] P. S. Hasmita and C. F. Sianturi, "Implementasi Algoritma Punctured Elias Code Untuk Kompresi File Audio Pada Aplikasi Lagu Rohani," vol. 10, pp. 46–50, 2021.
- [6] E. Siahaan, "Implementasi Algoritma Elias Gamma Code Untuk Kompresi File Teks Pada Aplikasi Watpad," *RESOLUSI : Rekayasa Teknik Informatika dan Informasi*, vol. 2, no. 2, pp. 76–84, 2021, [Online]. Available: <https://djournal.com/resolusi>
- [7] D. Cahayati, A. M. H. Pardede, and H. Khair, "Implementasi Algoritma Elias Gamma Kompresi Pada File Teks," *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, vol. 06, no. 01, pp. 159–166, 2022.
- [8] D. Pratiwi and T. Zebua, "ANALISIS PERBANDINGAN KINERJA ALGORITMA FIXED LENGTH BINARY ENCODING DAN ALGORITMA ELIAS GAMMA CODE DALAM KOMPRESI FILE TEKS," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 424–430, Nov. 2019, doi: 10.30865/komik.v3i1.1623.
- [9] H. Sartika and T. Zebua, "Perancangan Dan Implementasi Algoritma Elias Gamma Code Untuk Mengkompresi Record Database Pada Aplikasi Rangkuman Pengetahuan Umum Lengkap," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 3, no. 1, pp. 259–265, 2019, doi: 10.30865/komik.v3i1.1600.
- [10] David Salomon, "Penerapan kombinasi algoritma sequiter dan punctured elias code untuk kompresi file teks," vol. 2, no. 2, pp. 55–59, 2010.